Naval Research Laboratory

Washington, DC 20375-5000

NRL Report 9335

A Hierarchical Artificial Intelligence Maintenance Advisor

J. A. MOLNAR

Communications Systems Branch
Information Technology Division

G. J. MOSS, JR.

Naval Oceanographic and Atmospheric Research Laboratory
Stennis Space Center, MS

July 8, 1991

Approved for public release, distribution unlimited

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 8, 1991	Final (1989 to present)

4. TITLE AND SUBTITLE

A Hierarchical Artificial Intelligence Maintenance Advisor

5. FUNDING NUMBERS

WU-DN 680-125

6. AUTHOR(S)

J. A. Molnar and G. J. Moss, Jr.*

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL Report 9335

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Sea Systems Command
Washington, DC

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

*Naval Oceanographic and Atmospheric Research Laboratory
Stennis Space Center, MS

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Research was conducted to develop an artificial intelligence software package that assists technicians in maintenance of the Coded Pulse processor, a subsystem of the AN/SQS-53B. The AI software resides on a centrally located computer and is accessible via a terminal. An architecture was demonstrated consisting of a two layer hierarchy: Functional and Isolation. The circuit topology of the Coded Pulse processor provided the basis for knowledge acquisition. The hierarchy significantly reduces the processing time required for the large knowledge base.

The hierarchy replicates the system's fault isolation strategy. System monitoring information is entered by a technician into the Functional layer of the hierarchy. The Functional layer determines if the Coded Pulse processor is performing improperly and identifies the functional abnormality. Control is then directed to the Isolation layer. The Isolation layer directs the technician to perform the most efficient troubleshooting tests. Replacement recommendations are made based on adjustments made to the relative probability information during testing. Multiple faults can be identified and system testing can be redirected to other functional areas. Functional layer knowledge must be tailored to the system. Knowledge bases are required for each functional area of the Isolation layer. System architecture and results are discussed.

14. SUBJECT TERMS				15. NUMBER OF PAGES
Maintenance	Hierarchy	Artificial Intelligence	Analog	32
Expert System	Multiple faults	Sonar		16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

CONTENTS

A HIERARCHICAL ARTIFICIAL INTELLIGENCE MAINTENANCE ADVISOR

INTRODUCTION

The Navy's Ticonderoga-class cruiser fields an advanced assortment of electronic systems. This class of ship was specifically designed as the platform for the Aegis system. It has the latest developments in warfare systems for surface combatants. Included in the list of electronic systems (beginning with the CG-56) is the active sonar system AN/SQS-53B. The AN/SQS-53B uses 12 processing channels: 4 left, 4 center, and 4 right. Transmission and reception of acoustic signals are performed with mostly analog circuitry, which is predominantly used in the system front end. After detection, signals are processed by the Coded Pulse (CP) processor, where an analog to digital conversion is performed. Signals beyond this point in the system (or downstream in the circuit path) are processed digitally.

The AN/SQS-53B has the capability to monitor the system's internal performance at the functional level. Initial information of system performance is obtained from the display console, where each of the 12 channels has a display track on a single CRT. The information presented on the CRT relates the return signal to moving objects, stationary obstacles, clutter, noise, etc. Either an overabundance or absence of clutter and noise may provide the first indication that the system is malfunctioning. A built-in self-test subsystem, Performance Monitoring Fault Locator (PMFL), provides operational test information. Test generators are also used to present off-line or mode-specific test vectors to indicate system performance. The functional tests require a minimal level of effort by the operator, i.e., switch selection. Other tests are available as a secondary level of performance monitoring. These are tests that require cabinets to be opened so that indicator light responses may be noted. All are performance tests because they assess the normal functionality of the system. Each of the performance tests is linked to functions executed by the system.

Beyond performance tests, diagnostic tests are used to isolate the cause of an abnormality in a monitored function. Diagnostic tests are performed by trained technicians who monitor test responses with general purpose test equipment (GPTE) connected at system testpoints. Troubleshooting is based on training, experience, and technical manual information.

The Technician's Assister System (TAS), an Artificial Intelligence (AI) system, was developed to assist technicians in the maintenance process. A hierarchy was developed for TAS that uses the sonar system's internal monitoring capability to direct a technician to the proper functional area. Further assistance is supplied to the technician by the suggestion of diagnostic tests. The intended result leads the technician to a suggestion for card (module) replacement. Testing is directed along the most efficient path as determined by resultant information or relative test execution cost. Replacement is based on probabilistic reasoning.

Specifically, the TAS is a proof-of-concept model to address troubleshooting of the CP processor. The CP processor was selected because: digital and analog circuits are both present; gaps exist in PMFL coverage, requiring manual fault isolation; diagnostic testing is a time consuming process; it is sufficiently complex to challenge maintenance personnel conceptually; it is large enough to demonstrate the principle (over 100 replaceable modules); and it is small enough to demonstrate utility in fault isolation.

Manuscript approved April 5, 1991

The body of this report develops the overall concept of TAS and the Appendix, a graphic and textual presentation made to the Defense Preparedness Association, explains the use of the TAS software.

EVOLUTION OF THE ARCHITECTURE

The developed architecture arose from the desire to increase the speed of accessing information within the expert system. Initially, a single knowledge database (knowledge base) comprising 3600 rules was used to describe the entire electrical relationship of modules with the CP processor [1]. With a SUN Microsystems, Inc. Model 3/110 computer, the access time for retrieving information in the full CP processor model required 20 minutes in certain instances. Access time of this magnitude was deemed unacceptable for aiding technicians in the process of fault isolation. This led to the conclusion that a hierarchical architecture was necessary to reduce processing time, as well as to provide for eventual expansion of the troubleshooting domain to the sonar system or combat system level.

As a test of the hypothesis that a hierarchical architecture would reduce processing times to an acceptable level, a simplified trial case was created. A subset of the knowledge base was created to describe a single channel, channel seven. The single-channel knowledge base was compiled. Run time operations were then compared to similar operations conducted on the large knowledge base. Improvements were noted in the test recommendation and fault isolation times. On average, the test suggestion time range was $10 \text{ s} < t < 120 \text{ s}$. As a result, fault isolation was an order of magnitude faster for the single-channel trial case. Results from the single-channel knowledge partition suggested that the entire CP processor could be adequately described with a hierarchy that used channel symmetry and functional boundaries. Technician manipulation of obscure knowledge bases was perceived to be cumbersome and error laden. The solution chosen was to provide a two-layer hierarchy, with the initial layer encountered being the Local Area Expert (LAE) and the next layer being the Fault Isolation Layer (FIL). The LAE directs maintenance to a single specific functional area based on information obtained by the technician from system indicators and PMFL. The FIL is subordinate to the LAE procedurally but operationally is responsible for testing and maintenance procedures performed by the technician. Additionally, replacement modules are suggested by the FIL.

LAE

The LAE itself is an expert system whose primary function is to direct the technician to maintenance of a single functional area, and any additional functional areas until all malfunctions are removed. The LAE is not intended to isolate faults beyond the functional level.

The LAE performs the following tasks: initializing the knowledge state, updating knowledge, isolating to the functional area, invoking the FIL with the correct knowledge base, and passing the updated knowledge state to the FIL.

The LAE was constructed using C code generated by the Rulemaster expert system shell as a basis. The C code was then customized for efficiency and enhanced to facilitate integration with the FIL. Primarily, Rulemaster was used to incorporate performance-monitoring logic presented by the AN/UYQ-21 display characteristics, Receiver Test Generator (RTG), PMFL, and indicator lights. The Rulemaster-generated code was cumbersome, in terms of efficiency and storage space, and circuitous in logic explanations. This software was streamlined to include the basic logic but reduce the layered explanations and procedural steps used to achieve functional isolation.

Procedural steps relate the various species of performance tests (e.g., the monitor's display of channel tracks is grouped as a single indicator test). Explanations of the logical path were condensed from the broad restatement of a diagnostic path to concise statements relating to functionality. The result was a concise expression of the knowledge presented by the sonar system's built-in fault isolation architecture.

Once the LAE is invoked by typing "lae" at the UNIX prompt, the user is queried for whether the knowledge state of a previous session should be loaded or a new session should be initiated. A continuation of a previous maintenance session recovers the knowledge state from a file written when the session was halted. Primarily, sessions that are interrupted by priority operational tasks or that extend over work shifts are recovered.

New sessions begin with an examination of the PMFL indicators as displayed on the Support Monitoring Unit (Unit 14). Only the PMFLs relevant to the CP processor are displayed to the user. These are indicators 60-66, which correlate to power supply modules, and indicators 294-300, which correlate faults with the various reference signals in the sonar system. The display of any of these PMFL indicators leads to the secondary layer of indicator light interrogations, which identifies functional abnormalities. The result of identifying functional abnormalities leads, through heuristic reasoning, to the selection of a single functional area to fault-isolate.

The presence of a detected functional abnormality combined with absence of the associated PMFL fault code leads to another branch of logic. This branch considers the System Monitor as faulted and addresses functional isolation through other indicators such as console display and the Receiver Test Generator (RTG). If appropriate, it provides implications that supporting units outside of the CP processor may also be faulted, and suggests that these supporting units be repaired before making further investigations into the CP processor's functionality.

The result determined by the LAE states which function should be further fault-isolated. Initially, if the LAE determines that multiple functional areas are faulted, the heuristics select a single function to interrogate. At the point of isolation to a functional entity, the user is presented with a textual description of the achieved result and an explanation of its consequence. The explanation is presented as ADVICE to the user.

After the ADVICE presentation, the user is informed that the FIL is being invoked. Furthermore, the knowledge base corresponding to the functional area believed to be faulted is automatically loaded into computer memory.

From the first invocation of the FIL, the presence of the LAE is maintained only as a knowledge state tabulator. Primarily, the LAE function keeps track of which tests have been performed by the FIL, the result obtained, any subsequent invalidations that occurred, and the present function being investigated. The FIL interchanges this information with the LAE at the conclusion of a fault isolation process. The LAE continues to redirect fault isolation to other functional areas, but only as a result of conclusions reached by the FIL. In general, the LAE would only actively participate in the selection of the first functional area isolation; beyond that, the FIL should have sufficient general information to redirect troubleshooting. The LAE achieves the status of the functional area selector only if the FIL has insufficient knowledge to discriminate a new path (usually an indication of a knowledge hole or misconstruction of a functional area knowledge base).

Primarily, the LAE, under the normal conditions of well-constructed knowledge bases, achieves the role of CP processor functionality verifier. As such, the LAE presents all branches of functionality logic and requests the user to verify that malfunction indications are not present. Barring the presence of a residual malfunction, the LAE terminates its maintenance involvement. Residual malfunction states would be resolved by the FIL after the LAE has heuristically selected the function abnormality.

FIL

The FIL is based on the Fault Isolation System (FIS) Inference Engine [2, 3] developed at the Naval Research Laboratory (NRL). The FIL possesses enhancements of modified user interfaces, probability based heuristics (to recommend replacements), LAE integration software, and other subtle enhancements. The FIL's primary function is to isolate faults within a functional area to the replaceable module level. Component-level faults are not considered. Isolation to the component level is not part of the Navy's maintenance strategy. The process performed by the FIL is summarized as follows: the knowledge base is loaded into the FIL; the user interacts with the FIL to achieve an efficient troubleshooting sequence; and module replacement and redirection to other functional areas is then recommended. The FIL is capable of addressing both single and multiple module faults.

Currently, the FIL is written in LUCID COMMON LISP for the SUN 3/110. While variants of FIS have been written for other computers and compilers, FIL variants do not exist. The FIL consists of two primary parts: the inference engine and the knowledge base. The inference engine is based on probabilistic reasoning. Implications result from each user update of information. The implications are summed to lead to a preponderance of evidence. Thus, entry of faulty information, while not ideal, does not lead the FIL to pursue a path to incorrect module

isolation. Instead, faulty information may increase the possibility that a module may be incorrectly identified as a possible cause, but correct information will adjust the probability accordingly.

In addition to calculating the probability of the various modules being faulty, the inference engine determines the most efficient path to obtain new information, based on the current state of disorder in the system knowledge and relative cost of diagnostic tests. The FIL primarily chooses each step in the path based on which test yields the most knowledge on the state of the system (usually information on the greatest number of modules), or which test yields the most knowledge in the least costly manner. This primary selection factor is modified by the second factor, cost. Cost is provided as a priori knowledge and is based on the time required to perform a testing action. As a result, tests requiring extensive use of time are selected only if they will provide a relatively large increase in the knowledge state of the system.

All information used by the inference engine is either contained in the knowledge base or supplied by the technician as a result of a FIL request. Primarily, the knowledge base contains a priori knowledge of the physical structure of the unit under test (UUT). The knowledge base consists of six unique groups of information. The rules are essential, because they express the causal nature of system diagnostics in relation to the physical topology of the UUT's circuitry. Causal rules were generated directly from the schematics of the CP processor. Test information contained by the knowledge base relates physical measurements to logical rule structures. Precondition information is used to activate and deactivate the appropriate rules when switches are thrown or the configuration is changed. Order information is essential only if a group of tests must be performed in a certain sequence. Instruction information is presented in the knowledge base as a convenience for user interface and is optional. If instructions exist, an index relating them to tests must also be present.

In the system architecture, the functional areas identified in the LAE must relate to a knowledge base for each functional area. Thus, because the LAE can identify 12 unique functional areas for the CP processor, 12 different knowledge bases are used. Their functional unit names are: left correlator, right correlator, center correlator, left channels, right channels, center channels, PMFL, power, timing, multiplexer reference, modulator reference, and correlator reference. Overlap of the knowledge bases supports redirection of the testing effort. Each knowledge base represents adjacent functional areas as if they were replaceable modules; diagnosis of failure to a macromodule triggers redirection of the diagnostic effort to the appropriate knowledge base.

The structure of the FIL execution begins with the LAE invoking the FIL and passing it the functional isolation information. The information passed between layers is the chosen functional knowledge base to load and the available test result information. After the appropriate knowledge base is loaded, the test result information is used to update the system probability state. The initial system state consists of the module default failure probabilities obtained from the normalized reliability data. The probability state is then adjusted by performing the information update using the results passed by the LAE. All updating occurs before the interface is presented to the technician for continued fault isolation.

The interface presented after information updating is displayed in menu format categorized into three areas: control, inspection, and troubleshooting. Control provides commands to redirect the analysis to the LAE or to other functional areas. Output device control functions also exist. The inspection menu provides tools for assessing the state of the system probability state. Two primary functions allow the user to make judgments on likely faults from their output: show probabilities and ambiguity set. The show probabilities item provides a list of modules with their respective probabilities of being faulty and certified. Probabilities of being faulty are accumulated from the result of a failed test and weighted by the logical proximity of the module to the testpoint. Modules in the ambiguity set of a faulted test have their fault probability raised. Other modules may not have their fault probability impacted or may have it lowered. Certification is the result of a weighted probability based on the total number of module tests and the logical proximity of the test to the module. Certification is accumulated by all modules in the ambiguity set of the test; other modules do not gain certification from the test result. A condensed presentation of the 10 most probable module failures is also available in bar graph form.

The ambiguity set command is available to inspect the modules in the ambiguity set of the last faulted test. This command, together with the probability information, gives the user the capability to make judgment

replacements at any point based on the information presented and experience. Other functions are available in the inspection menu to display the list of tests that have been performed.

The troubleshooting menu is used most often because it directs and selects testing through best-test and make-test commands. The ability to remove the result of a test is also available. The best-test command promotes efficient testing, based on the heuristic drive to increase system "knowledge" and cost. The user has the option of accepting the recommendation or specifying a different test using the make-test command. For reference, instructions are presented to the user. The user performs the physical test in either case and enters the result. After each testing action, the probabilities, certifications, and system probability state are recalculated.

Testing continues until some form of resolution is reached. One possibility is for the user to choose to make a replacement based on the information available and experience. Another case is that the system isolates a fault to a module or modules, and the technician performs the replacement. The final possibility is that the FIL isolates the fault to a macromodule, a condensed logical representation of the knowledge base for another functional area. In the first two cases, the user has the option to continue troubleshooting (usually to confirm functionality) or to return to the LAE. In the last possibility, troubleshooting is redirected to the other function's knowledge base by loading the new knowledge base and then continuing troubleshooting. All of the cases involve the interaction of the LAE and FIL at an interface.

INTERACTION OF LAE AND FIL

The interaction of the two layers can simply be described as follows: the LAE is used to perform isolation to the functional area, and the FIL is used to perform fault isolation to the replaceable module. Figure 1 indicates the flow of control. Figure 1 and the LAE section of this report show that the flow of the fault isolation process begins with the sonar system technician detecting an error state in the CP processor. The LAE is invoked and queries the state of system fault codes. Isolation is made to the functional level. The LAE then invokes the FIL and passes two information items: the name of the functional knowledge base that should be loaded and a list of test information obtained.

The FIL loads the suggested knowledge base and initializes the system state with the known test results. The FIL then continues the testing at the module level by assisting the technician in choosing the most efficient tests. Testing continues until the technician opts to replace a module by using experience and the information obtained from the FIL, or the FIL automatically suggests a replacement.

When a decision is made to perform a replacement, those test results related to the module replaced (i.e., those with the module in their ambiguity set) are invalidated. Tests whose validity does not depend on the module replacement remain valid. The FIL passes all the information on valid and invalid tests to the next cycle of fault isolation.

The next cycle of isolation has two different paths depending on the type of replacement. If a single or multiple module replacement is made, control is returned to the LAE to affirm the functionality of the CP processor. If the LAE confirms the functionality of the CP processor, the system exits to UNIX. If additional functional errors are found by the LAE, the cycle begins again. The second path is taken if a macromodule is suggested for replacement. In this case, the LAE updates its database of exercised test results and reinvokes the FIL with the appropriate knowledge base. The cyclic reasoning continues until all malfunctions in the CP processor are eliminated.

VALIDATION

Validation of the TAS occurred over several stages in the development. Initially, the knowledge base was correlated with the schematic diagrams to account for inaccuracies in the causal logic. Upon compilation of the knowledge base into the FIL form, diagnostics were presented for the detection of syntax and gross format errors. After compilation, utilities to examine rule and ambiguity consistencies were used to examine the knowledge linkage. Finally, prototype simulations were executed to test the convergence of the knowledge bases.

After a great deal of confidence in the knowledge integrity was obtained, prototype field testing was performed at the Naval Underwater Systems Center (NUSC) AN/SQS-53B test bed in New London, CT. One particular fault was diagnosed during the demonstration of the prototype TAS - a fault in a delay line of the CP processor. While full testing and validation of the TAS remain before final implementation, we feel that the prototype demonstrated a high degree of feasibility for application to a total troubleshooting architecture.

CONCLUSIONS

A two layer architecture was demonstrated to address the fault isolation requirements for the CP processor of the AN/SQS-53B. The architecture incorporated one layer to perform functional isolation and another layer to perform isolation to faulty replaceable modules. The prototype exhibited both logical consistency and the capability to isolate physical faults during the validation and testing phases. The architecture is generic, which allows it to be extended to the entire sonar system or even beyond to several ship systems. The main requirement for specific heuristic information at the functional level of isolation (LAE) is viewed as the only tailoring required to extend the complexity coverage of the system.

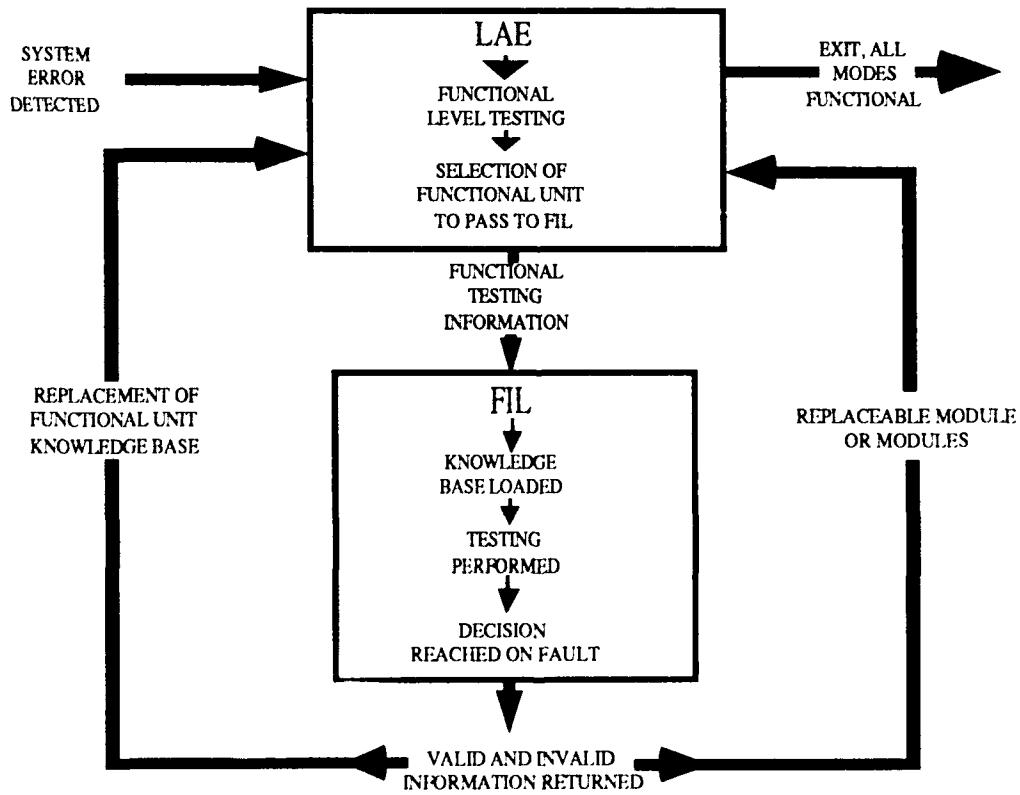


Fig. 1 — Representation of the TAS hierarchy, displaying the major functions of the LAE, FIL and interface process

REFERENCES

1. J. A. Molnar, "Creation, Validation, Testing, and Data Management of a Knowledge Base Designed for a Technician's Assister System for the AN/SQS-53B, Unit 26, Using a Fault Isolation System Shell," NRL Report 9296, Dec. 1990.

2. F. Pipitone, "The FIS Electronics Troubleshooting System," *IEEE Computer Magazine* 19(7), 68-76 (1986).

3. F. Pipitone, K. A. DeJong, and W. Spears, "An Artificial Intelligence Approach to Analog Systems Diagnosis," NRL Report 9219, Sept. 1989.

ACKNOWLEDGMENTS

We thank the Naval Sea Systems Command, PMS411, for their support and interest in addressing the problem of electronic system maintenance through the use of artificial intelligence techniques; Dr. Frank Pipitone and Mr. William Spears for their advice; and Dr. John Davis, Dr. Randall Shumaker, Mr. Lee Kline, and Mr. Thomas Gattis for their enthusiasm for the project.

Appendix

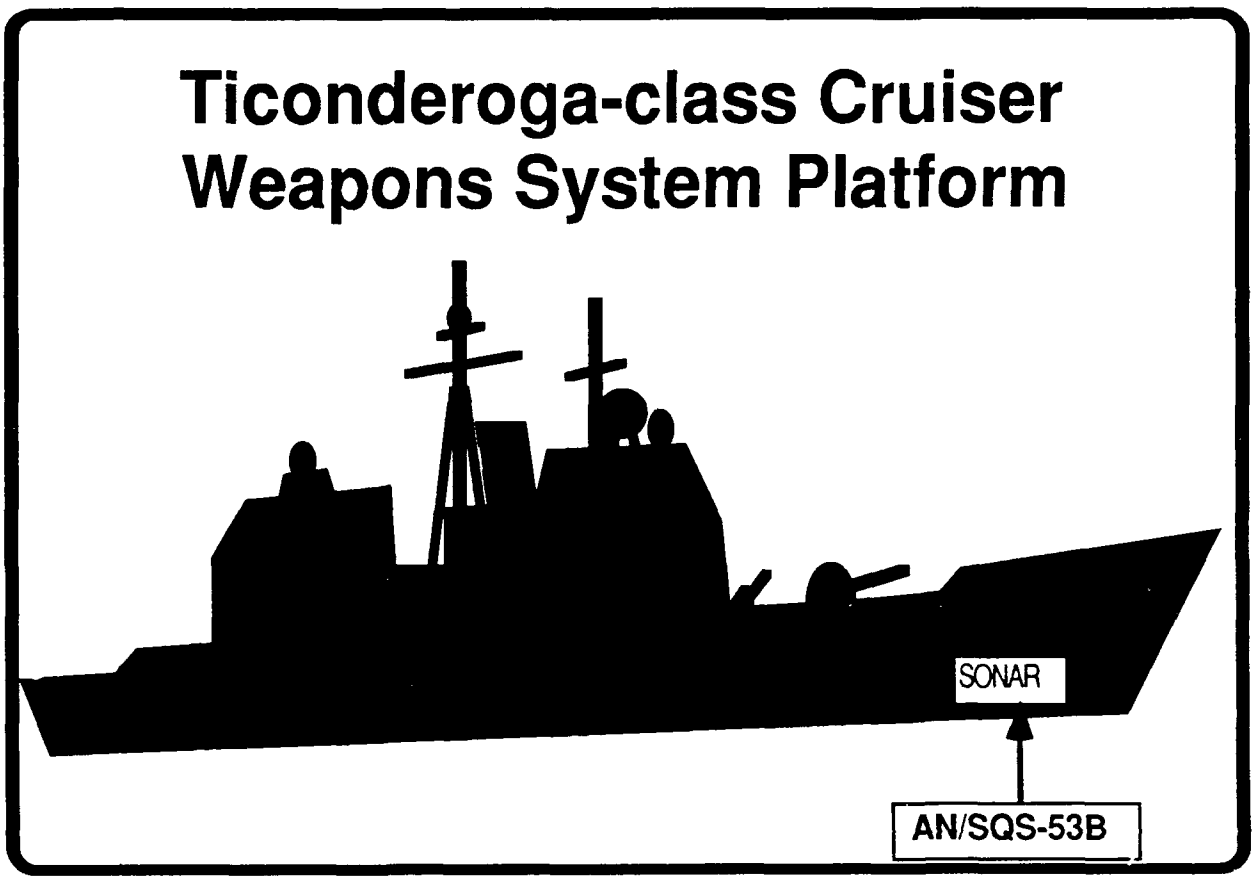
ARTIFICIAL INTELLIGENCE APPLICATIONS FOR MILITARY LOGISTICS

This Appendix contains the visuals and summarizes commentary presented at the Defense Preparedness Association conference on "Artificial Intelligence Applications for Military Logistics," Williamsburg, VA, March 27-30, 1990.

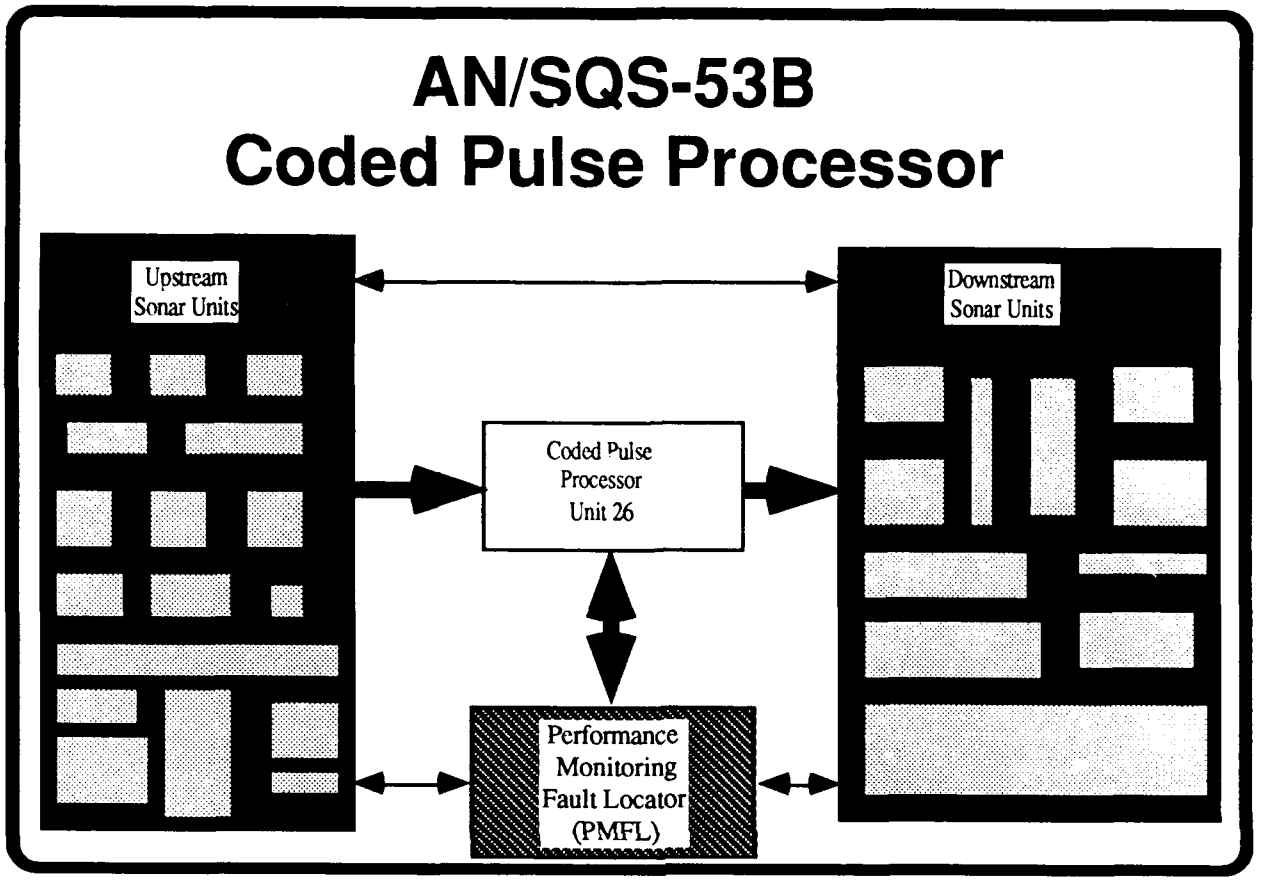
The presentation, "A Hierarchical Artificial Intelligence Maintenance Advisor," was written by Joseph A. Molnar (NRL) and George J. Moss, Jr. (NOARL). Mr. Molnar made the presentation.

The research presented was the result of a joint effort pursued by the authors during the 3-year period of 1987 through 1989. The effort was sponsored and supported by the Naval Sea Systems Command, PMS 411.

The following presentation summarizes the research performed to develop an expert system to aid a technician in performing electronic maintenance troubleshooting on the Unit 26 subsystem of the AN/SQS-53B sonar system.

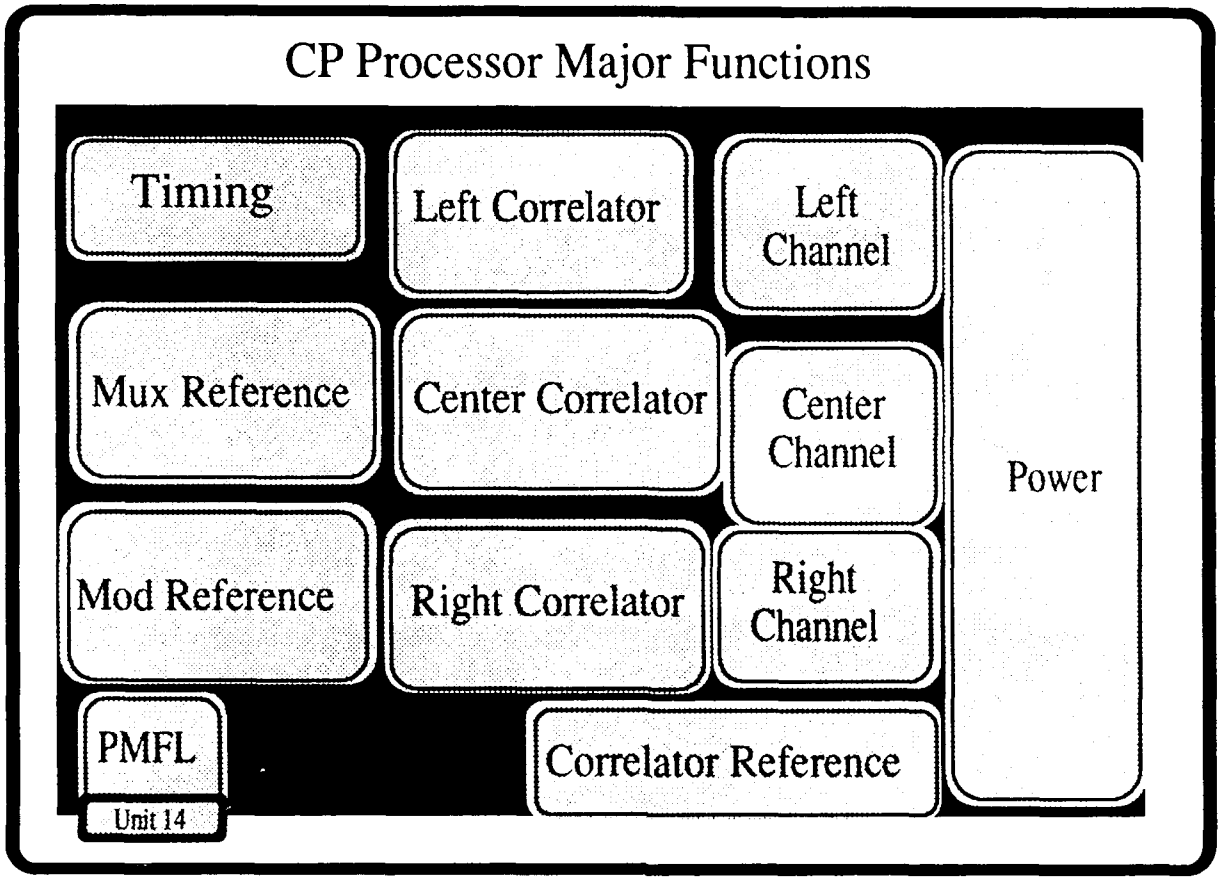


The AN/SQS-53B is the hull mounted sonar system implemented on the Ticonderoga-class cruiser. It has subsystems distributed throughout the ship. Because its subsystems use both analog and digital processing techniques, the AN/SQS-53B was chosen to demonstrate the feasibility of developing an expert system to aid a technician in repairing electronic subsystems. Furthermore, Unit 26, the Coded Pulse processor (a representative subsystem containing both analog and digital circuits), was selected for the demonstration because of its complexity and reliability.



Unit 26 was modeled as a subsystem that received signals from other units, such as Upstream Sonar Units (the signals are operated on by these units before they reach Unit 26), and transmitted signals to other units, such as Downstream sonar units.

The Upstream and Downstream units comprise several actual equipment subsystems similar to Unit 26, but for the purposes of the model they are grouped and represented as a single unit. This is represented in the diagram as the patterned rectangles embedded within the black rectangles representing the Upstream and Downstream Sonar Units. Individual relationships within these units are not defined.



Unit 26 comprises circuitry that performs 12 functions. One function, the PMFL, directly relates to a portion of the functionality of Unit 14, the PMFL unit. The operation of several Unit 26 circuit modules is shared by various functions. For example, the modules composing the power function actively influence the operation of all other functions. As another example, a delay line module may have signals from both the left and right correlator functions. Thus, while the functions are displayed in the diagram as distinct, the circuitry modules used to perform that function are shared.

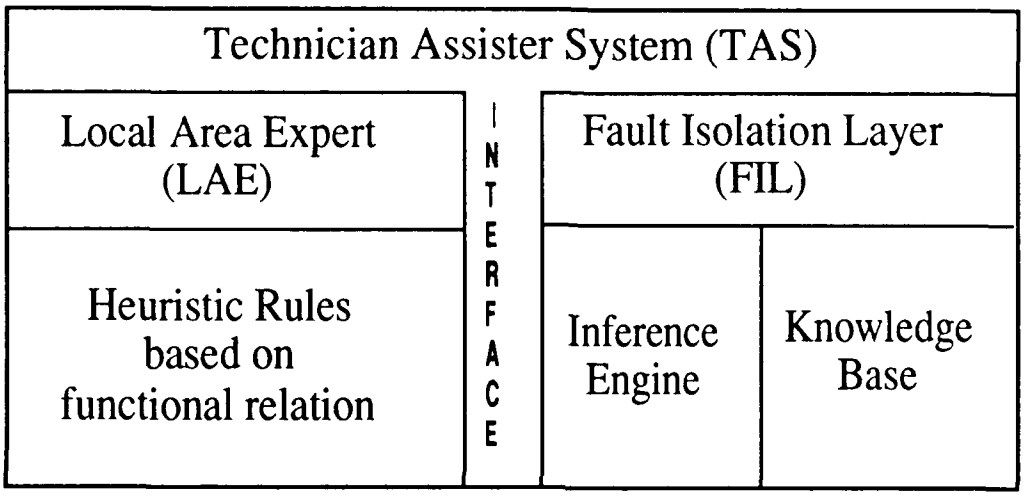
Electronic Composition of Unit 26

- 120 replaceable modules or PC boards
- Modules have overlapping or multiple functionality
- Over 600 defined testpoints

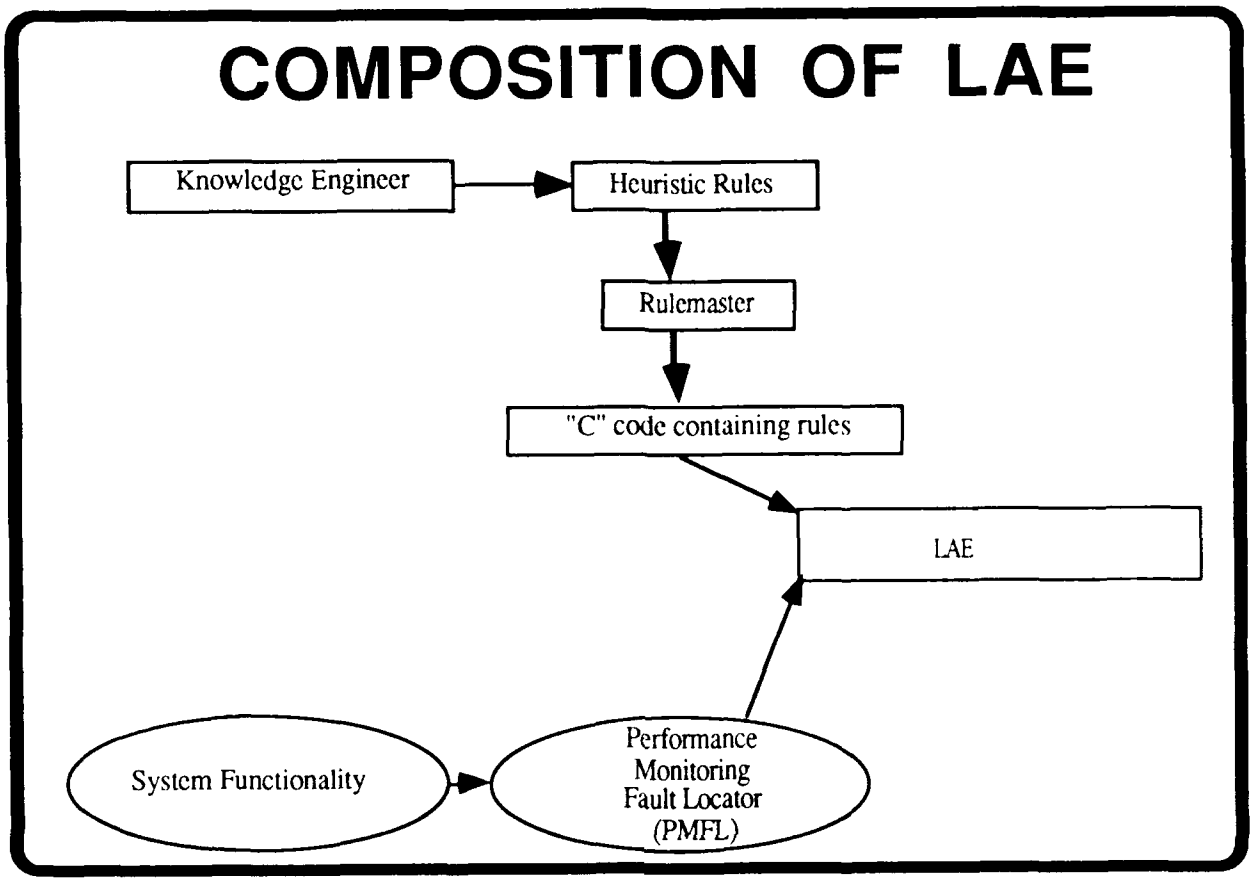
The circuitry of Unit 26 is contained on printed circuit (PC) boards. The maintenance philosophy of the system identifies the PC boards as replaceable modules. Thus, maintenance on the system is complete at the operational level when failed modules are identified and replaced. Identification is performed by the system technician using system technical manuals and general-purpose electronic test equipment. Training and technical manuals define the tests that the technician may perform and how they are executed. The symptoms exhibited by the system define the order in which technicians perform those tests.

The expert system was developed to assist the technician in selecting the next appropriate test based on the information obtained from previous tests. Additionally, it presents the technician with procedural information on how to perform the test. The expert system was designed to assist technicians with all levels of experience and, additionally, to act as a training device.

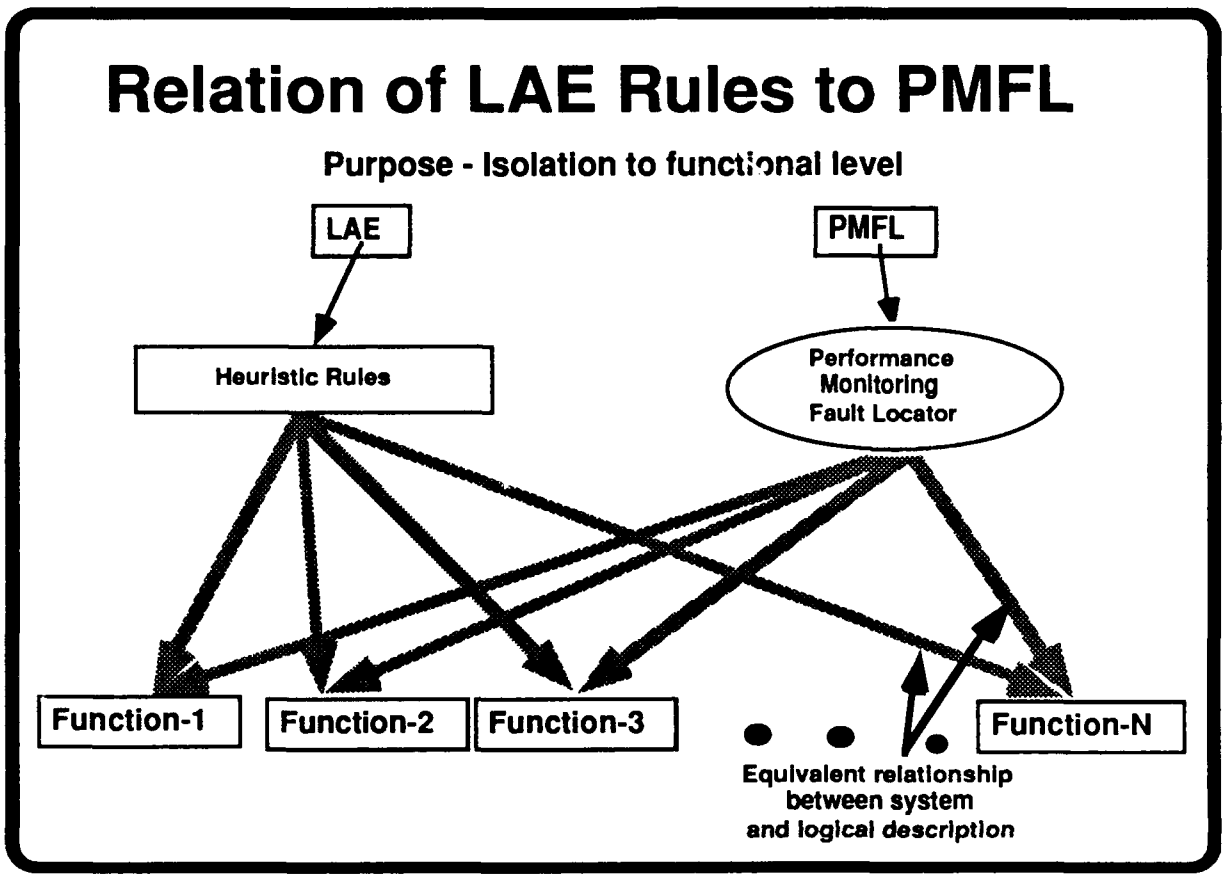
Introduction to Technician Assister System Hierarchy



The expert system developed is named the Technician Assister System (TAS). Essentially, the TAS is composed of two separate layers, a Local Area Expert (LAE) and a Fault Isolation Layer (FIL). The TAS provides the interface for interaction between these layers. The LAE comprises a heuristic expert system to identify functional faults within the AN/SQS-53B that relate to malfunctions in Unit 26. The FIL is a model-based expert system that relates malfunctions in the Unit 26 subsystem to faults in the circuit modules.



The LAE was created by a knowledge engineer who developed the heuristic rules. The rules were based on the functionality of the sonar system. Functional errors in the sonar system are identified by the PMFL. The heuristic rules express the relationship between the system functionality and the PMFL's identification of errors. The heuristic rules were entered into the Rulemaster expert system shell. Rulemaster was then used to generate the C source code that became the LAE, which is the executable program module.



The LAE mirrors the operation of the PMFL to create the logical relationship between the system hardware and the system function performed by the hardware. Each test performed by the PMFL provides the operation status of the function performed by that circuit. The LAE replicates the testing process to determine for TAS the logical model of system Unit 26 functions that correlates with fault indications. If several functions are faulted, the LAE acts to direct further inspection of the system.

LAE as Knowledge Manager

- **Accounts for the validity of all tests**

- **Passes information between FIL actions**

The LAE manages the information transfer as FIL acts to isolate module failure. Specifically, after each maintenance action, the LAE maintains the status of valid knowledge obtained through testing. It also presents this knowledge state to FIL each time it is invoked.

Composition of FIL

Inference Engine

- Developed by NCARAI
- Use best test heuristics to suggest testing sequence
- Produces probabilistic representation of system state

Knowledge Base

- Causal rules
- Description of circuit topology
- Logical representation of physical interconnections
- Knowledge base relation to system function

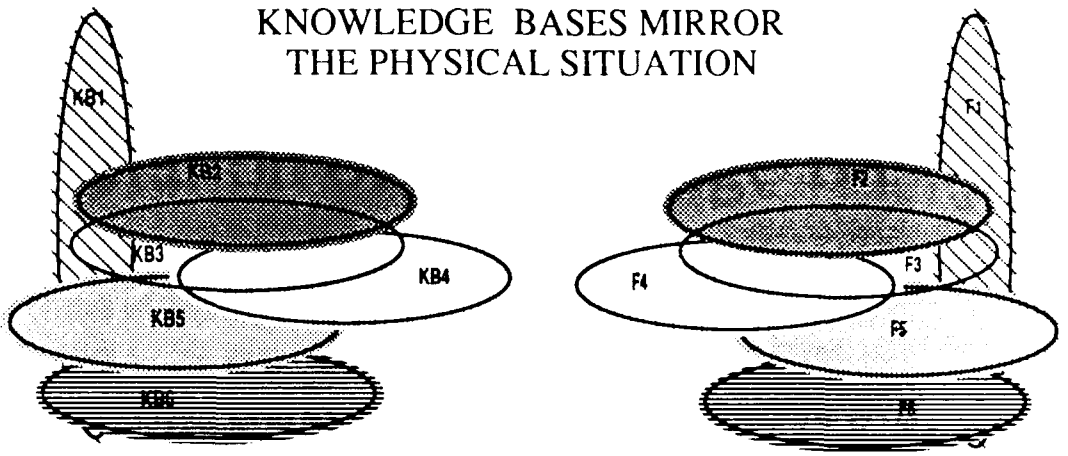
The FIL uses the Fault Isolation System (FIS) developed at the Navy Center for Applied Research in Artificial Intelligence (NCARAI). Modifications were performed to integrate the software as a module in the TAS hierarchy.

The FIL comprises two distinct parts, the Inference Engine and the Knowledge Base. The Inference Engine is a shell that acts on the model of the circuit topology that is expressed in the knowledge base. The Inference Engine uses heuristics to determine the testing sequence of the electronic system. The sequence is modified based on the results of previous tests. Each test result modifies the knowledge state, expressed in probabilities, of the system.

The knowledge base comprises causal rules that form a logical representation of electrical signal flow among modules. The knowledge base contains other information that relates the physical system to the logical model (i.e., tests, instructions, preconditions, and orders). These are used by the FIL in conjunction with the causal rules to identify the physical tests that can be performed on Unit 26, provide information on how to perform the test, allow FIL to optimize the testing sequence based on preconditions that must exist to perform the test, and define for FIL any mandatory testing sequences.

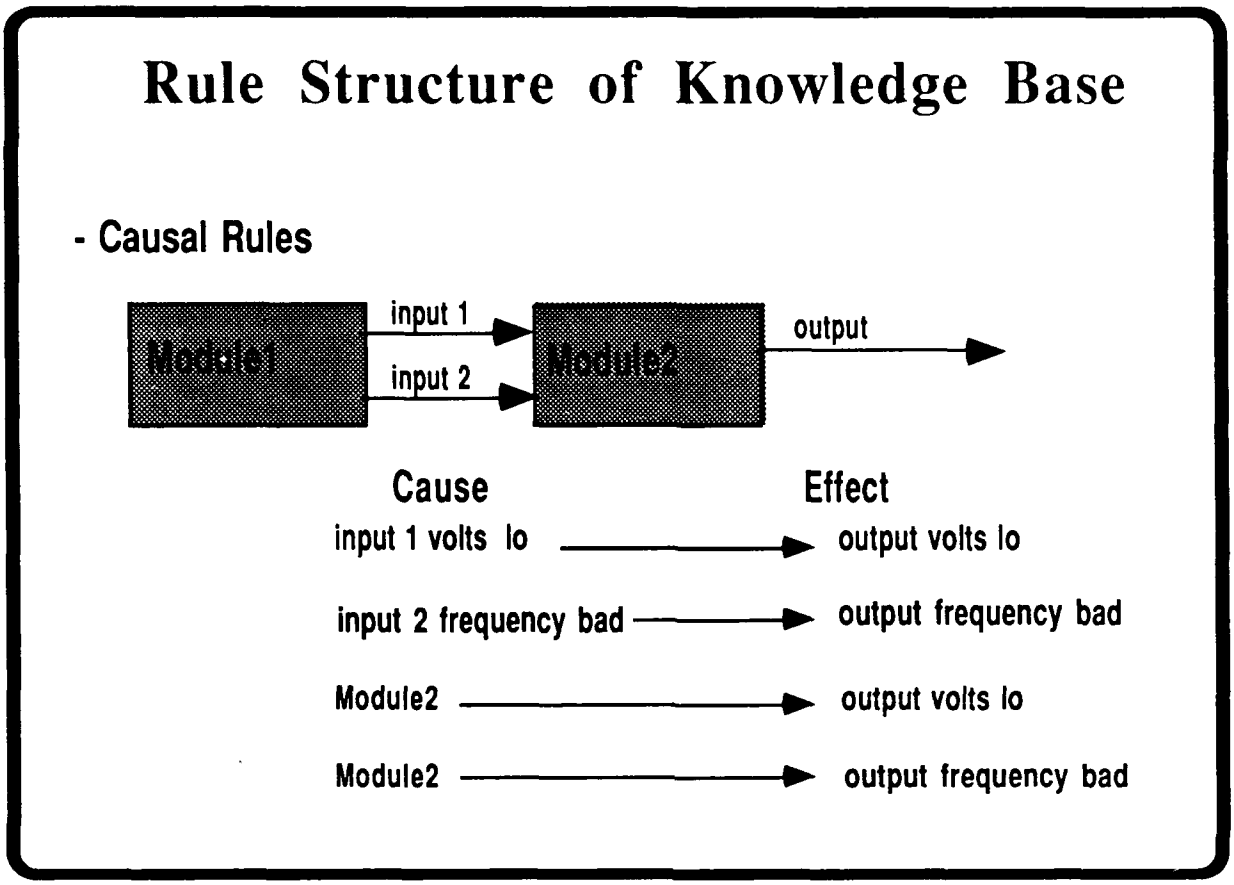
Relation of Knowledge Base

- **A Knowledge base exists for each functional unit described in LAE (total of 12)**
- **Functional Units overlap at the replaceable module level**
- **Knowledge bases overlap also at the replaceable module level**

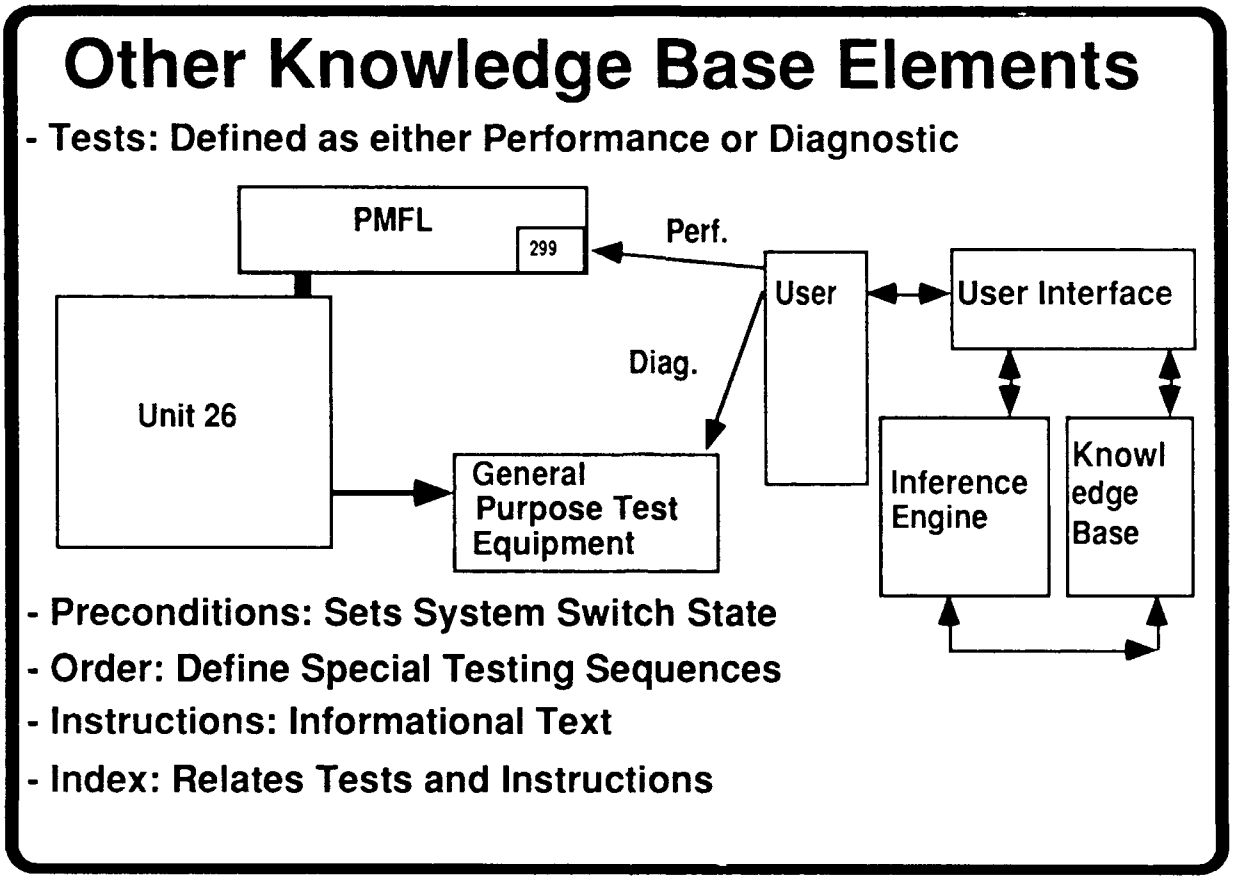


The knowledge bases of the FIL were created to mirror the functionality of Unit 26. As an example, the visual displays six functions, F1 - F6, that represent the relationship between the modules that perform the various functions. The knowledge bases that model the interrelationships of the modules in functions F1 - F6 are displayed as KB1 - KB6. The functionality of the system is mirrored in the example of the knowledge base representation.

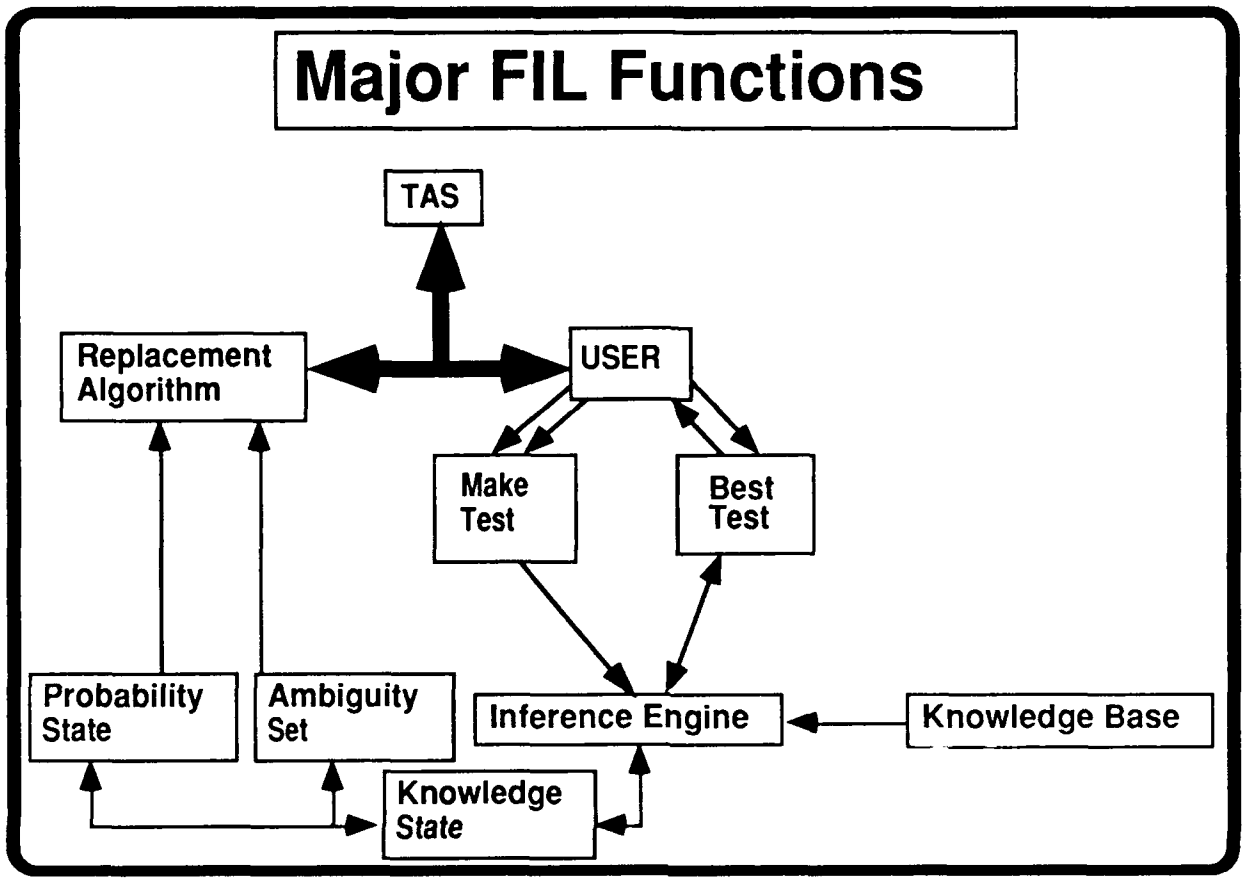
As stated earlier, Unit 26 comprises 12 functions. A knowledge base was created for each of the 12 functions. The modules used to perform the functions of Unit 26 are represented as modules in the knowledge base. As in the physical system, the logical representation of modules may appear in several knowledge bases since they perform a similar task for several functions. This is represented by the overlapping illustrated in the diagram. For each knowledge base, functions that interact are represented as macromodules. A macromodule is a condensed logical representation indicating the connection relation between modules in one functional unit and those in another functional unit. For example: Knowledge Base 2 (KB2) defines all physical modules that are used to perform the function (F2) as logical modules. Modules used to perform multiple functions are described in multiple knowledge bases - in this case, modules of KB1, KB3, and KB4. Finally, a relation is defined that relates a module in KB2 to a path to modules in another knowledge base, say KB1. The paths to multiple modules are represented as a single path to a module representing the group of modules contained in KB1. This module that represents the group is called a macromodule. The macromodule is given the name of the knowledge base where the modules are modeled. In this manner, the physical relationship to other knowledge bases is contained in the macromodule representation.



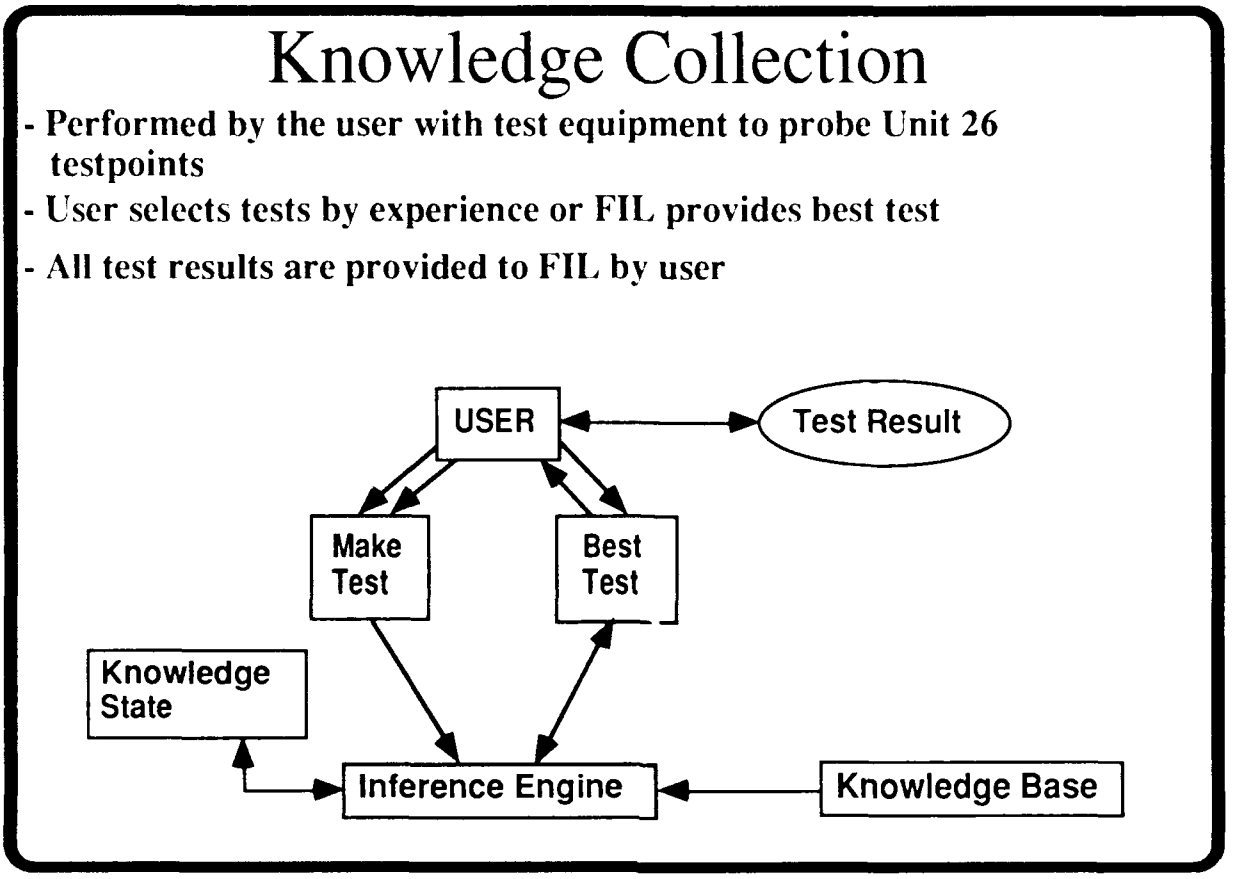
The causal rules of the knowledge base are described from the physical relationship of modules. In the diagram, Module1 and Module2 are replaceable modules. Input1, Input2, and Output are terminals of the modules. Causal rules are created by defining two cases: (1) an input signal to a module can result in a faulted output signal, and (2) a fault within a module can result in a faulted output signal. The effect in either case is defined by three distinct parts: the terminal where the output is occurring, the parameter affected, and the condition of the parameter. In the first case, the cause is described by a three part expression. The first part of the expression defines the terminal location. The second part of the expression defines the parameter that would be measured at that terminal. The final part of the expression describes the condition at the particular terminal for a parameter. The expression for the effect follows the same form as the three-part expression for the cause. In the second case, the cause is described in a single term (i.e., the module name). Sets of rules are constructed in this manner for each module.



This diagram indicates the interaction that must occur between the sonar system, expert system, and user. The user is guided by the expert system in making tests. The tests are either performance or diagnostic. Performance tests are obtained by the user while viewing the PMFL display. Diagnostic tests are performed by the user while using test equipment to perform physical measurements on the system at defined testpoints. In either case, the user must supply the test results through the user interface of the TAS, which is hosted on a SUN workstation.



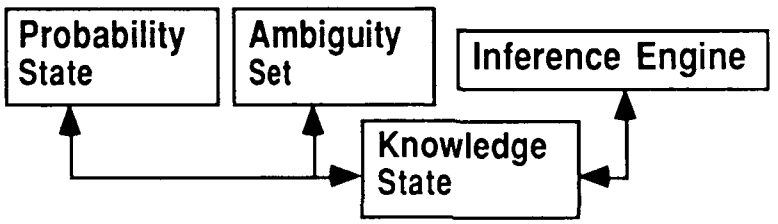
This diagram illustrates the major functions of the FIL. The user's first interactions are through the interface to select a test sequence. If the user selects the best test function, FIL provides the user with the test that is most efficient, considering results from the prior history of tests. Users may also select a function to make a test. In this case, the user must then supply the test selection. In general, the make test function would be used if the technician has a belief on what the problem may be and wishes to test that hypothesis. The result is supplied by the user after performing the test.



Results are acted on by the inference engine. The inference engine updates the knowledge state based on the results. The updated knowledge state of the system is used when acting upon the knowledge base created from the model of the hardware function. When the user then requests a new best test, the inference engine operates on the knowledge base to again provide the most efficient test based on the most recently updated knowledge state.

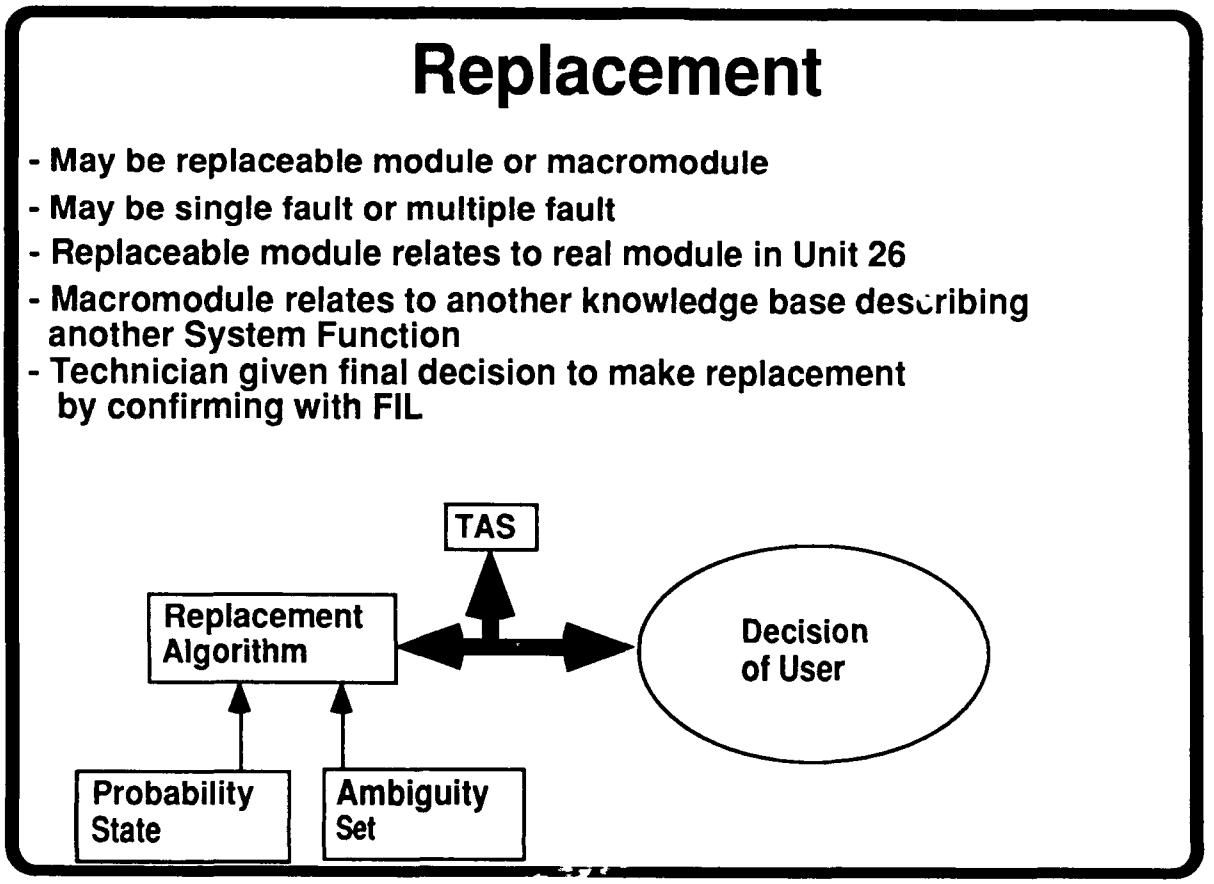
Knowledge State Update

- As a result of every test result FIL adjusts its beliefs
- Changes to Probability state
- Failed tests adjust the Ambiguity set
- Failed tests add to belief that modules in ambiguity set are faulty
- Passed tests add to belief that modules in ambiguity set are functional

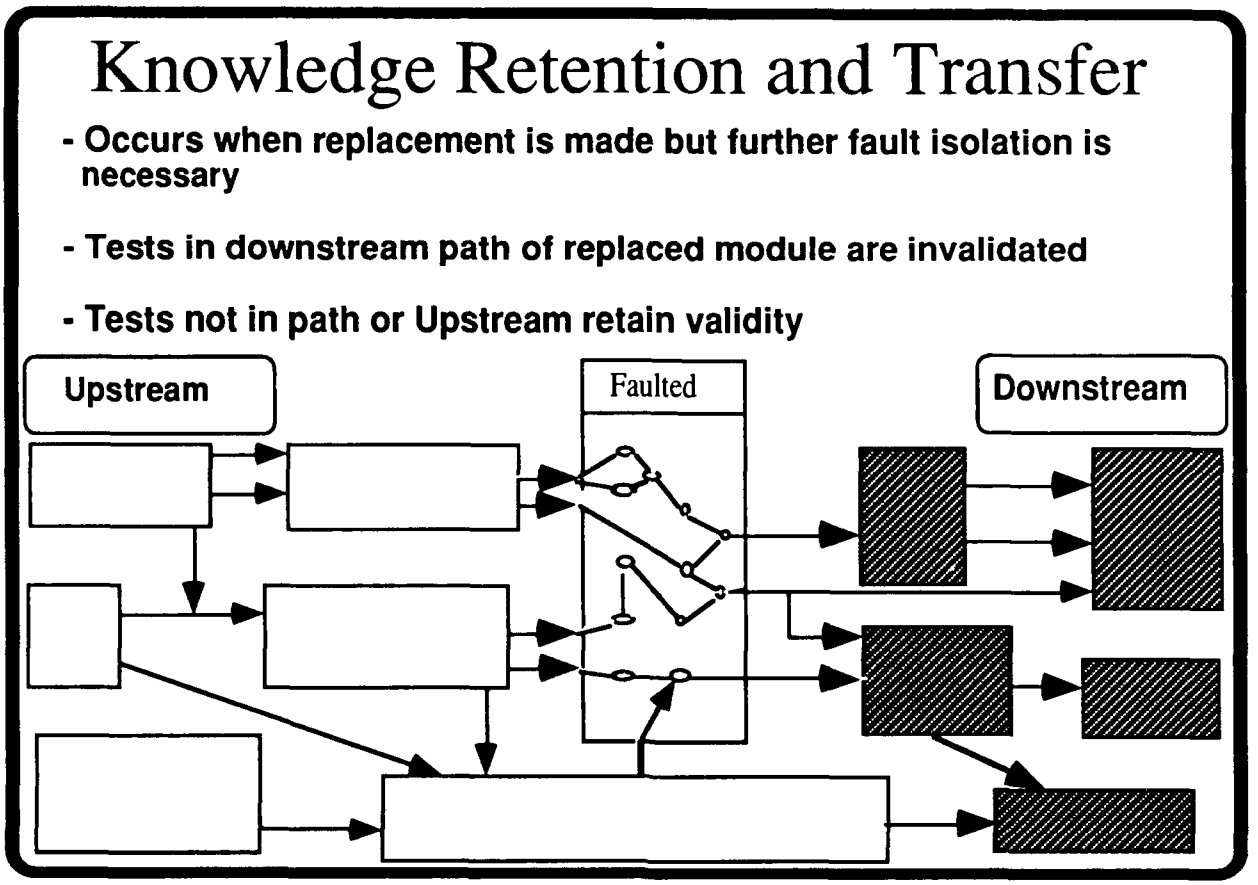


A result of updating the knowledge state of the system results in changing the probability state and the ambiguity set. The probability state consists of individual probabilities that each module is malfunctioning. A certification is also associated with the probability state. The certification provides a quantitative assessment of what proportion of tests for that module has passed. The probability of all modules will change as a result of either a passed or failed test depending upon the module's logical relation and distance from the test. Certification is only altered when tests are passed.

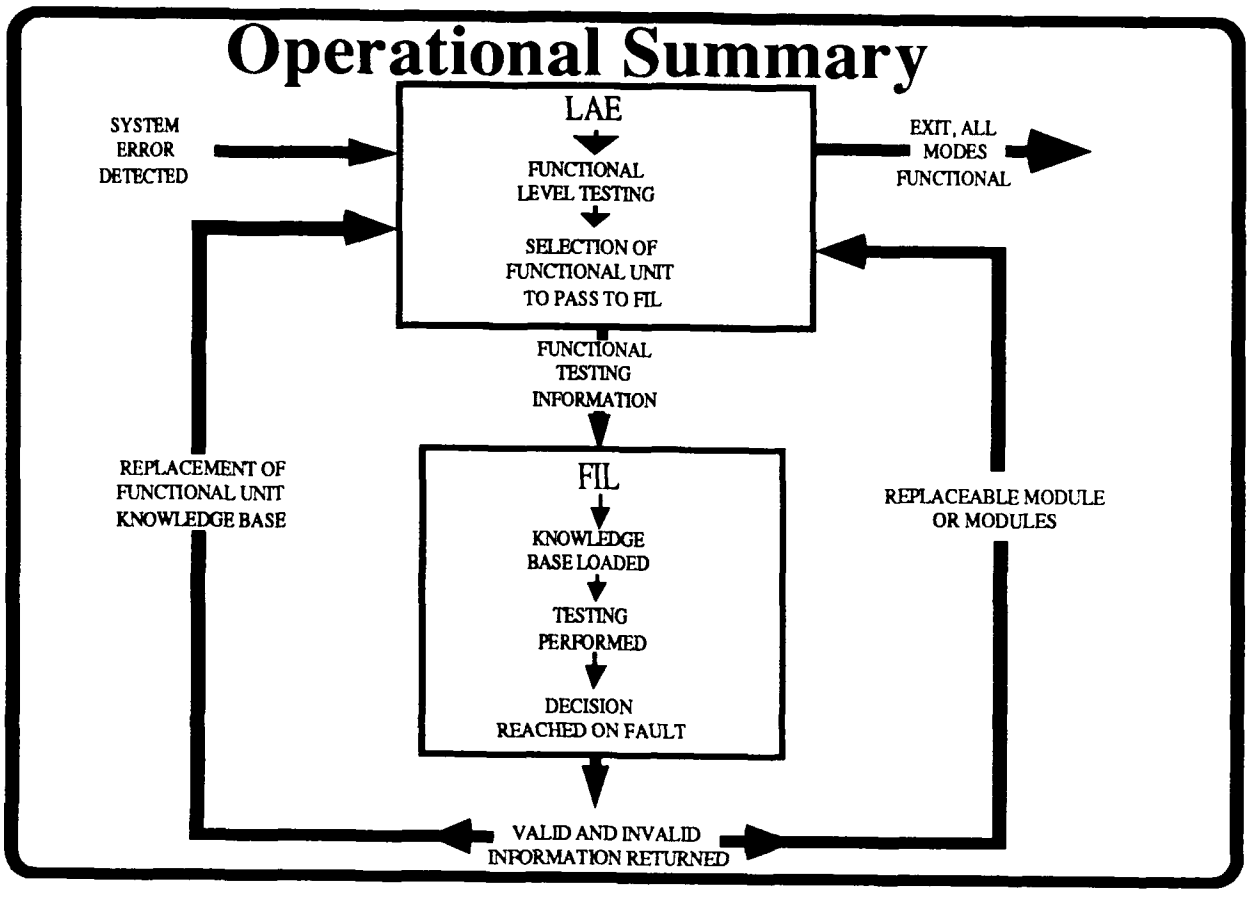
The ambiguity set is adjusted after every failed test to correspond with modules in the ambiguity set of the test. The ambiguity set is not altered if a test is passed. The FIL performs essential acts to reduce the size of the ambiguity set.



As a result of the changes in the probability and ambiguity information, a replacement algorithm uses thresholds to alert the user to likely module faults. The replacements indicated may be either single or multiple faults. The faults may be either replaceable modules or macromodules. If the user agrees to the replacement of a macromodule, the FIL will inform the TAS and replace the current knowledge base with the knowledge base corresponding to the macromodule name. At this point, the FIL will act upon this new knowledge base when assigning a knowledge state.



As replacements are made, the results of tests downstream of the module replacement become invalid. As the diagram indicates, the module in the center is faulted and must be replaced. When a new module is provided, it is obvious that the results of tests made on the faulted module are no longer valid. Furthermore, because signals that passed through the faulted module were also corrupted, tests made in the downstream path, the patterned modules, are also invalid. The FIL invalidates all appropriate tests and maintains the validity of tests performed on upstream modules. If the TAS, through interaction with the LAE, determines additional faults are present, then the FIL already has a preset knowledge state arising from the previous valid tests. The object is to maintain the knowledge state at the highest possible level. Should the last replacement be a macromodule, all tests remain valid since only the knowledge base is replaced for FIL.



The Operational Summary provides a flow of decisions in compact form. The objective is that when a system error is detected, the LAE first is used to determine the functional area faulted. At that point, all of the information obtained, as well as operation control, is transferred to the FIL. The FIL loads into computer memory the indicated knowledge base and then recommends further detailed testing. At some point, thresholds will be surpassed and module replacements will be recommended.

If the module recommended for replacement is a macromodule, all of the previous test results will be valid. At this point, the FIL has determined that another functional area of the sonar system contains the fault. If the user confirms that the replacement of the macromodule is made, the FIL will load the new knowledge base corresponding to the macromodule. Testing continues as before until the faults are located.

If the module recommended for replacement is a replaceable module, as opposed to a macromodule, then some of the results from the preceding test sequence will not be valid. Upon the user's confirmation of the replacement, appropriate test results are invalidated. Other tests retain their validity as testing proceeds. The LAE then determines if all malfunctions are eliminated. If additional malfunctions are found, a similar cycle is repeated until all malfunctions are eliminated. At this point, the sonar system should be fully operational, at least for the functions of Unit 26. The TAS is then exited.

Conclusions

- Prototype developed to logically describe Unit 26
- Consists of LAE, heuristic rules, and FIL; 12 knowledge bases to describe major functions
- Verified knowledge
- Validation suggested by earlier experiment with Unit 26
- TAS test suggestion times on the order of 1 minute or less

The following are the results of this development.

A prototype was developed to accurately describe the Unit 26 subsystem of the AN/SQS-53B.

A hierarchy was developed that uses two layers. One layer, the LAE, provides fault isolation to the functional level. The LAE is based on heuristic rules. The other layer, the FIL, isolates faults to the replaceable module level. The FIL is a model-based expert system. The hierarchy is compatible with extending the TAS to provide a similar capability for other modules in the system.

The knowledge bases were verified using utilities contained in the FIL.

Validation was suggested by an earlier experiment that used only the FIL layer and a single knowledge base to accurately identify fault within that functional area of the sonar system. Testing was performed at the sonar system test bed at the Naval Underwater Systems Center, New London, CT.

Test suggestions provided by the FIL layer of the TAS took a maximum of 1 minute to obtain. This was compatible with technicians working in an operational environment and did not provide a burdensome time delay. The majority of tests provided were much faster (approximately 1 - 2 seconds).

Our research project provided favorable results, and it appears that further development and/or extension to additional parts of the sonar system would be successful. The techniques developed in this project would be applicable by others who wish to develop a similar system to aid in the fault isolation process of other complete electronic systems that use a mixture of analog and digital components.